IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Field et al.

Application No.: 09/540,576
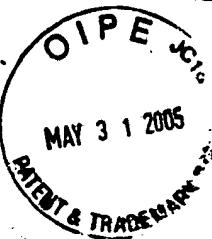
Filed: March 31, 2000

Title: DEBUGGER PROTOCOL GENERATOR

Attorney Docket No.: SUN1P252/P4198

Examiner: Kiss, Eric B.

Group: 2122

Confirmation No.: 2536

DECLARATION (B) UNDER 37 C.F.R. § 1.132

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

I, Robert Field, declare the following:

1. I am one of the inventors of the invention set forth in the above referenced application.

2. I gave a presentation entitled "The New Java™ Platform Debugger Architecture," on March 26, 1998, at the Moscone Convention Center in San Francisco.

3. Gordon Hirsch, the other co-inventor of the invention set forth in the above referenced application, also participated in the presentation entitled "The New Java™ Platform Debugger Architecture," on March 26, 1998, at the Moscone Convention Center in San Francisco.

4. I was present during the entire presentation entitled "The New Java™ Platform Debugger Architecture," on March 26, 1998, at the Moscone Convention Center in San Francisco.

5. I have reviewed the claims of the above-identified application (claims 20 and 24-32).

6. The concept of "automatically generating a front-end and a back end debugger program based on parsing of a formal specification" (which is recited, for example, in claim 20 of the above referenced application) was NOT publicly disclosed or otherwise made available to the public

during the presentation entitled "The New Java™ Platform Debugger Architecture," on March 26, 1998, at the Moscone Convention Center in San Francisco.

7. I have reviewed the "Old Jbug schedule" document which details the formal testing schedule for the Java™ Platform Debugger Architecture (Jbug), (a copy of which is enclosed herewith).

8. The entry marked as "JDWP generation" in the "Old Jbug schedule" document corresponds to the concept of "automatically generating a front-end and a back-end debugger program based on parsing of a formal specification," and the "Early Access" release of the Java™ Platform Debugger Architecture (Jbug) is represented by the column marked as "EA" in the Old Jbug schedule" document.

9. As the "Old Jbug schedule" clearly indicates, the column marked as "EA" which indicates software development activity leading up to the "Early Access" experimental testing period of the Java™ Platform Debugger Architecture (Jbug) does NOT have an entry for the "JDWP generation" feature, thus showing that the May 1998 "Early access" of the Java™ Platform Debugger Architecture (Jbug) did NOT include the concept of "automatically generating a front-end and a back-end debugger program based on parsing of a formal specification."

10. I have also reviewed the provisional application 60/145, 136, filed on July 21, 1999.

11. I note that the provisional application 60/145, 136, filed on July 21, 1999 includes a section entitled the "changes since early access" (a copy of which is attached herewith) which clearly indicates that "JDWP now formally specified and mechanically generated."

12. The statement: "JDWP now formally specified and mechanically generated" in the section entitled the "changes since early access" of the provisional application 60/145, 136, filed on July 21, 1999 refers to the concept of "automatically generating a front-end and a back-end debugger program based on parsing of a formal specification."

13. Based on my review of the "Old Jbug schedule" and the provisional application 60/145, 136, filed July 21, 1999, I am confident that the concept of "automatically generating a front-end and a back-end debugger program based on parsing of a formal specification" was NOT released in the May 1998 "Early access" of the Java™ Platform Debugger Architecture (Jbug). Nor had internal development of this even begun.

14. As indicated by the "old Jbug schedule" document, it is my belief and understanding that software development and subsequent experimental testing of "JDWP generation" started on September 1998. Such development was performed personally by me (as indicated by my first initial 'R' in the 'Who' column, see the 'Key' at the bottom) and subsequent testing was conducted under my direct supervision and control.

15. I have also reviewed the Jbug-beta Software License (a copy of which is attached herewith).

16. As indicated by the Jbug-beta Software License, both Beta 1 and Beta 2 testing of the Java[TM] Platform Debugger Architecture (Jbug) were experimental test releases.

17. The Beta 1 and Beta 2 test releases of the Jbug test software were performed under my direct control and supervision in accordance with the limitations and restrictions set forth in the Jbug Beta Software License for a sixty (60) day period (please see, Jbug-beta Software License, section 1, BETA TESTING PERIOD) in order to fix any defects or deficiencies that were present in the experimental version of the Java[TM] Platform Debugger Architecture (Jbug).

18. I have reviewed test-reports we received back from the testers during Beta 1 and Beta 2 testing of the Java[TM] Platform Debugger Architecture (Jbug) (three (3) sample emails are submitted herewith).

19. As can be seen from the sample emails of the test-reports noted above (paragraph 20), input was solicited and received from testers of the Java[TM] Platform Debugger Architecture (Jbug) at least until June 1, 1999.

20. The Java[TM] platform Debugger Architecture (Jbug) was modified during the Beta 1 and Beta 2 experimental testing period based on test-reports we received back from the testers.

21. The Jbug Beta Software License includes a non-disclosure agreement that strictly forbids disclosure of confidential information by the testers to any third party (Jbug Beta Software License, section 3).

22. I have no reason to believe that the non-disclosure agreement of the Jbug Beta Software License was violated by any of the testers during the Beta 1 or Beta 2 testing.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true. I further declare that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both (under Section 1001 of Title 18 of the United States Code), and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date May 24, 2005   Robert Field:

BEST AVAILABLE COPY
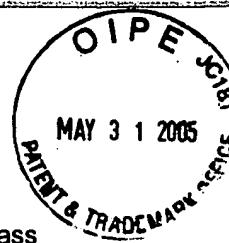
From:	Rajeswari <rraji@siptech.co.in>
To:	<rfield@Eng>
Date:	3/2/1999 4:55:27 AM
Subject:	Clarification in com.sun.jdi.Method class

Hi Robert,

While testing the method returnType() in com.sun.jdi.Method I am getting
an exception . Please clarify this point.
I'm sending the testcase for this method.

Class - com.sun.jdi.Method
Method - returnType()

Testing of method returnType() in class Method gives the following
error.
----
java.lang.IllegalArgumentException: Unrecognized signature V
        at
com.sun.tools.jdi.PrimitiveTypeImpl.getTypeMirror(PrimitiveTypeImpl.j
ava:56)
        at
com.sun.tools.jdi.ReferenceTypeImpl.findType(ReferenceTypeImpl.java:6
64)
        at com.sun.tools.jdi.MethodImpl.returnType(MethodImpl.java:365)
        at jdi.tests.MethodTest1.testJDIReturnType(MethodTest1.java:35)
        at java.lang.reflect.Method.invoke(Native Method)
        at jdi.tests.JDITestCase.runTest(JDITestCase.java:32)
        at test.framework.TestCase.run(TestCase.java:230)
        at test.framework.TestSuite.run(TestSuite.java:96)
        at jdi.tests.JDITestSuite.run(JDITestSuite.java:24)
        at jdi.tests.TestAll.run(TestAll.java:137)
        at jdi.tests.TestAll.main(TestAll.java:96)

After jbug testsuite execution the above error occurs. Is this the
correct behaviour .

The method that is used does not return anything .It is in
jbug/test/src/share/classes/jdi/tests/program/MainClass.java

```
public void print(OutputStream out) {
 String string = fInt++ + ". " + fString;
 System.out.println(string);
}
```

Attachments :

MethodTest1.java
TestAll.java

Testcase - jbug/test/src/share/classes/jdi/tests/MethodTest1.java
Line
    classes.addElement(MethodTest1.class);
is added in getAllTestCases() method of class

jbug/test/src/share/classes/jdi/tests/TestAll.java

thanks
Raji

CC: Ajay Athavale <Ajay.Athavale@Eng>, <charath@siptech.co.in>, <sbalu@siptech.co.in>, "jtools_memb@siptech.co.in" <jtools_memb@siptech.co.in>

From:	Rob Morgan <rmorgan@silverstream.com>
To:	<java-debugger@java.sun.com>
Date:	3/4/1999 11:28:12 AM
Subject:	Problems with Jbug and Hotspot

I'm sure you guys know about these already but ...

1) With the Hotspot RC1 Win32 build installed, Jbug stops working.
2) Even if you start TTY with the -classic option you're still hosed
because the child process that is launched uses Hotspot.

This begs a couple of questions ...

Will the 2 VMs be transparently interchangable for debugging (ie.
debugger running under Hotspot can debug a process running under the
classic VM, and vice-versa)? It would kind of suck if they weren't. And,
what's the timeframe for getting Jbug to work with Hotspot?

- rob

From: Robert Field <robert.field@eng.sun.com>
To: Gordon Hirsch <gordon.hirsch@sun.com>
Date: 6/1/1999 2:32:57 PM
**Subject:**   Re: Clarifications to JDWP spec

Gordon Hirsch wrote:
> ...
> Alan Bateman - JavaSoft wrote:
> >...
> > 1. In the StackFrame command set Insignia reported an undocumented
> > byte in the reply to the 'ThisObject' command. I checked our
> > implementation and this is the Tag Constant that appears before
> > the 'objectID' in the reply. This should be documented in the
> > JDWP spec.
> >
> > 2. Insignia claim there is an undocumented byte prior to the
> > objectID in the EXCEPTION event. I was unable to diagnose what
> > they mean as they didn't supply an example. However I suspect they
> > refer to the Tag Constant again -- would I be correct to say that any
> > 'objectID' will be prefixed with a Tag Constant? If so then I think
> > that a number of places in the JDWP need to be updated.
>
> The current implementation has any object ID that is part of a reply
> prepended with the tag byte. Note that this is not the case for the more
> specific IDs like thread ID nor for Object IDs sent as part of commands.
> We'll need to decide whether to change this as it is a little inconsistent
> with other quantities we send back and forth over the wire.

The distinction between tagged and untagged objectIDs was made in the
formal spec and thus the generated code. The generated docs did not
make this distinction, a fix for this was just putback and will be in
the next build.

Please thank them for the report,
Robert

CC: Alan Bateman - JavaSoft <Alan.Bateman@ireland.sun.com>,
<java-debugger@eng.sun.com>
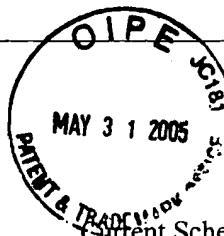
Current Schedule (beyond Beta)

Old Jbug Schedule (up to Beta)

Task	Days	Who	Apr	May	Jun	EA	Sep	Beta
EA								
reflection								
socket transport								
solaris								
breakpoints								
frames & variables								
exceptions								
step into								
thread info								
thread events								
link events								
start/stop world								
JDB on JDI								
thread control								
JDWP spec								
Beta								
thread groups								
step over								
monitor lock info (JVMDI)								
class redefinition interface								
watchpoints								
deferred watchpoints (try)								
reference mgmt								
method invoke								
finish stepping support								
filtering								
JDWP Generation	10							
Pluggable connectors and transports								
Rid dependency on Sun JVM interface								
GUI								
JVMDI spec update								
JDI use missing implementation								
Doc and release makefiles								

Does update	B	R					
Add method entry / exit to JDWP & JDI		R					
Add (sys)classpath to JDWP & JDI		R					
Add all locators (class, method) in JDWP & JDI		R					
Add Void value		R					
Factor expression parsing out of ttydebug		R					
Get rid of interrupted VM mode	S	G					

Key	
started	
done	
Robert Field	
Gordon Hirsch	
William "Bill" Maddox	
Alan Sommerer	
James McIlree	

November 16, 1998 rfield@eng.sun.com

Old Schedule (upto Beta)

Jbug FCS Schedule

Task	Days	Who	Beta	Beta2	FCS	1.1
FCS						
bug fixing	30	R&G		.		
Filter class loads	1	GR	++			
shared memory transport (Win32)	5	G		***		
Cache maintenance	5	G				
Add toString() wherever necessary	2	G		***		
Make JDI work with multiple VMs	3	G		***		
Improve stepping support by using METHOD_ENTER	3	G		***		
Fix ObjectReference.enableCollection/disableCollection	1	G		***		
Make ConnectorArgument serializable	1	G		***		
Allow "address=" ... in -Xrunjdwp	2	G		***		
Fix reporting of thread status	2	G		***		
Fix ThreadReference.stop for threads at breakpoints	1	G		***		
Allow a debugger to attach, detach, and attach again	5	G				
Combine co-located JVMDI events into JDWP event sets	1	G				
Switch to alternative malloc	2	G				
security audit	5	R&G				
polish JDWP & JDI specs	5	RG		.		
Polish GUI	20	R		.		
Docs update	5	R				
Futures						
expressions: back-end eval/conditions	20	R	.			
Call stack deltas	3	R				
Determine argument types/values when no debug info	3	R				
JDWP back-end event queue flush/priorities	5	G				
JDWP back-end event queue flush/priorities	5	G				
Complete SPI	10	G				
Solaris shared memory transport	3	G				
In process JDI	30	R&G				

Key (See Ownership)

	not done
.	started
++	mostly done
***	done
R	Robert Field **rfield@eng**
G	Gordon Hirsch **gordonh@eng**
AA	Ajay Aathavale (SQE) **ajay.aathavale@Eng**
AS	Alan Sommerer (Tech Pubs) **alans@eng**
-	Task not assigned
@	Current project

Event	When	What
Beta	December 23, 1998	final specs -> JDC binaries of Beta -> JDC source of Beta -> JDT
FCS	first quarter 1999 circa Cricket	JDI, JDWP & Jbug GUI -> bundled JDK

Last updated: January 11, 1999 *rfield@eng.sun.com*